T U T O R I A L

"Numerical Methods for the Solution of Elliptic Partial Differential Equations"

to the lecture

"Numerics of Elliptic Problems"

Tutorial 9 Tuesday, 30 May 2017, Time: $10^{\underline{15}}$ – $11^{\underline{45}}$, Room: HT 177F.

Programming (continued)

Incorporating boundary conditions

Consider the Neumann boundary value problem

$$\begin{aligned} -\Delta u(x) + u &= f(x) \qquad \text{for } x \in \Omega := (0,1)^2, \\ \frac{\partial u}{\partial n}(x) &= g(x) \qquad \text{for } x \in \Gamma_N := \partial\Omega\,. \end{aligned}$$

The associated variational formulation is to find $u \in V_0 := H^1(\Omega)$ such that

$$\int_\Omega \nabla u(x) \cdot \nabla v(x) + u(x)\, v(x)\, dx \;=\; \int_\Omega f(x)\, v(x)\, dx + \int_{\Gamma_N} g(x)\, v(x)\, ds \qquad \forall v \in V_0\,. \tag{3.30}$$

44 Let $e \subset \Gamma_N$ be an element edge on the Neumann boundary with the two endpoints $x^{(e,1)}$ and $x^{(e,2)}$ and set $h_e := |x^{(e,2)} - x^{(e,1)}|$. Let us denote the two functions on the reference edge by $p^{(1)}(\xi) = 1 - \xi$ and $p^{(2)}(\xi) = \xi$.

Write a function

```
void calcNeumannElVec (const Point2D& p0, const Point2D& p1,
                       ScalarField g, Vec<2>& elVec);
```

to approximate

$$g_e^{(\alpha)} := \int_e g(x)\, p^{(e,\alpha)}(x)\, ds \;\approx\; \frac{h_e}{2}\Big(g(x^{(e,1)})\, p^{(\alpha)}(0) + g(x^{(e,2)})\, p^{(\alpha)}(1)\Big)$$

as above by the trapezoidal rule; `elVec`$\approx (g_e^{(1)}, g_e^{(2)})$, `p0`$=x^{(e,1)}$, `p1`$=x^{(e,2)}$, and `g`$=g$.

45 Write a function

```
void addNeumannLoadVector (const Mesh& mesh, ScalarField g, Vector& b);
```

which *adds* the contribution corresponding to $\int_{\Gamma_N} g(x)\, v(x)\, ds$ to an (already existing) load vector `b`.

Hint: Loop over all segments of the mesh and for those marked as Neumann (use `bcSegments[i] == BC_NEUMANN`) call `calcNeumannElVec`.

46 Solve the finite element system corresponding to (3.30) with $f(x_1, x_2) = -2.5 + x_1$ and $g(x_1, x_2) = 0.5$ for a suitably refined mesh (see exercise **39**) and visualize the solution.

Consider the Dirichlet boundary value problem

$$\begin{aligned} -\Delta u(x) &= f(x) && \text{for } x \in \Omega := (0,1)^2, \\ u(x) &= g && \text{for } x \in \Gamma_D := \partial\Omega. \end{aligned}$$

The associated variational formulation is to find $u \in V_g := \{u \in H^1(\Omega) : u|_\Gamma = g\}$ such that

$$\int_\Omega \nabla u(x) \cdot \nabla v(x)\, dx = \int_\Omega f(x)\, v(x)\, dx \qquad \forall v \in V_g\,. \tag{3.31}$$

47 Write a function

```
void incorporateHomogeneousDirichletBC (const Mesh& mesh,
                                        SparseMatrix& K, Vector& b);
```

that incorporates the homogeneous Dirichlet boundary conditions ($g = 0$) into the system matrix `K` and the load vector `b`.

Hint: Loop over all segments of the mesh and search for those marked as Dirichlet (use `bcSegments[i] == BC_DIRICHLET`). For each such vertex with index i it sets all entries in row i and column i of K to zero and $K_{i,i} = 1$, $b_i = 0$.

48 Solve the finite element system corresponding to (3.31) with $f(x_1, x_2) = 20\pi^2 \sin(2\pi x_1) \sin(4\pi x_2)$ for a suitably refined mesh (see exercise **39**) and visualize the solution.

49 Write a function

```
void incorporateInhomogeneousDirichletBC (const Mesh& mesh,
                        const Vector& ug, SparseMatrix& K, Vector& b);
```

that incorporates the inhomogenenous Dirichlet boundary conditions `ug` into the system matrix `K` and the load vector `b`. Here `ug` is a vector of the same size as `b` carrying the prescribed Dirichlet values (other values are ignored).

Hint: Ensure that the entries in `ug`, that do not correspond to Dirichlet values are set to zero. The modification of the load vector `b` can be done by

$$\mathtt{b}[i] = \begin{cases} \mathtt{ug}[i], & i \text{ corresponds to Dirichlet node} \\ b[i] - (\mathtt{K} * \mathtt{ug})[i], & \text{else} \end{cases}$$

After that, in order to modify `K`, proceed as in Exercise **47**.

50 Solve the finite element system corresponding to (3.31) with $f(x_1, x_2) = 20\pi^2 \sin(2\pi x_1)\sin(4\pi x_2)$ and $g(x_1, x_2)$ given by

$$g(x_1, x_2) = \begin{cases} 0, & x_2 = 1 \vee x_1 = 1 \\ (1 - x_1), & x_2 = 0 \\ (1 - x_2), & x_1 = 0 \end{cases}$$

for a suitably refined mesh (see exercise **41**) and visualize the solution.

Let's consider Robin boundary conditions of the type

$$\frac{\partial u}{\partial N} := \lambda \frac{\partial u}{\partial n} = \kappa(u_0 - u) = g_3 - \kappa u.$$

for given λ, κ and u_0 and the normal derivative n.

51 Let $e \subset \Gamma_R$ be element edges on the Robin boundary with the two endpoints $x^{(e,1)}$ and $x^{(e,2)}$. Let the reference edge be $\Delta = (0, 1)$ with the corresponding nodal basis functions $p^{(0)}(\xi) = 1 - \xi$ and $p^{(1)}(\xi) = \xi$. Write a function

```
void calcRobinElMat (const Vec<2>& x0, const Vec<2>& x1,
                     ScalarField kappa, Mat<2, 2>& elMat);
```

that computes the element Robin matrix K

$$K^e_{\alpha\beta} = \int_e \kappa(x) p^{(e,\alpha)}(x) p^{(e,\beta)}(x) dx = \int_\Delta \kappa(x_e(\xi)) p^{(\alpha)}(\xi) p^{(\beta)}(\xi) \det(J_e) d\xi$$

using the quadrature rule on $\Delta = (0, 1)$ given by

$$\int_\Delta g(\xi) d\xi \approx \frac{1}{6}\left[g(0) + 4g(0.5) + g(1)\right].$$

Show that this quadrature rule is exact for $g \in P_3$.

Hint: In order to get $x_e(\xi)$, implement a class modelling the affine linear transformation for edges , i.e. in 1D (compare **31**, **32** and NumPDE-Tutorial).

52 Write a function

```
void incorporateRobinBC (const Mesh& mesh, ScalarField kappa,
                         ScalarField u0, SparseMatrix& K, Vector& b);
```

that incorporates the Robin boundary conditions into the system matrix `K` and the load vector `b`.

Hint: Loop over all segments of the mesh and search for those marked as Robin (use `bcSegments[i] == BC_ROBIN`) and reuse the function from the previous Exercise 51 to add the local contributions to the stiffness matrix.

Hint: For the contribution corresponding to $\int_{\Gamma_R} g_3(x)\, v(x)\, ds$, proceed as for the Neumann Boundary (see 44).